AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Currently amended) A computer-implemented method of scheduling a plurality of instructions generated by a compiler based on an intermediate representation of source code, the method comprising:

for each of one or more of the plurality of instructions of one or more instruction types ready to be scheduled in a given cycle in a scheduling region, determining a new slack value based on ~~current resource constraints, for each of one or more ready instructions in a scheduling region~~ a current maximum number of the instructions that can be scheduled in the given cycle for a target processor;

selecting ~~one of the ready instructions~~ up to the current maximum number of instructions from those instructions ready to be scheduled in the given cycle, based on a priority order associated with the new slack value~~ of the one ready instruction~~; and

scheduling the selected ~~ready~~ instructions~~; and~~

~~repeating the method for determining, selecting and scheduling for each of the one or more ready instructions remaining to be selected and scheduled until all ready instructions have been scheduled~~.

2. (Canceled)

3. (Currently amended) The method of claim 1, wherein determining the new slack value comprises:

determining the new slack value ~~for each of the one or more ready instructions~~

based additionally on resource constraints and dependence height for each of the one or more instructions.

4. (Currently amended) The method of claim ~~1~~ 3, wherein determining the new slack value comprises:

determining a dependence deadline based on [[a]] the dependence height for each of the one or more ~~ready~~ instructions;

determining a resource deadline based on the resource constraints for each of the one or more ~~ready~~ instructions;

selecting as a deadline value that indicates a least number of cycles, between the resource deadline and the dependence deadline; and

determining the new slack value based on the selected deadline value.

5. (Currently amended) The method of claim 1, wherein ~~selecting one of the ready instructions~~ the priority order comprises ~~selecting the ready~~ an ordering of the one or more instructions based on their ~~having a lowest~~ new slack value, wherein the instructions with lowest slack value have highest priority.

6. (Currently amended) The method of claim 1, further comprising:

generating an entry in a ready list for each of the one or more ~~ready~~ instructions prior to determining the new slack value; and

removing the entry for the selected ~~ready~~ instructions from the ready list responsive to scheduling the selected instructions.

7. (Currently amended) The method of claim 6, further comprising:

adding to an uncover list, any ~~non-ready~~ of the plurality of instructions not ready to be scheduled in the given cycle in the scheduling region that are uncovered by the scheduling of the selected ~~ready~~ instructions, wherein the ~~non-ready~~ instructions not ready to be scheduled are dependent on the selected ~~ready~~ instructions.

8. (Currently amended) The method of claim 6, wherein the given cycle is a given clock cycle, further comprising:

advancing a virtual clock to a subsequent clock cycle when there are no ~~ready~~ instructions in the ready list that can be scheduled in [[a]] the given clock cycle; and

adding an entry to the ready list for any ~~non-ready~~ of the plurality of instructions not ready to be scheduled in the given cycle in the scheduling region that becomes ready in the subsequent clock cycle.

9. (Currently amended) The method of claim 4, wherein determining the new slack value comprises:

determining a minimum number of cycles needed to schedule each of the one or more ~~ready~~ instructions in the scheduling region, taking the resource constraints into account;

determining the dependence deadline based on the dependence height and the minimum number of cycles; and

determining the resource deadline based on the resource constraints and the minimum number of cycles.

10. (Previously presented) The method of claim 9, wherein determining the minimum number of cycles comprises:

determining a dependence length of the scheduling region;

determining a resource length of the scheduling region;

assigning the dependence length as the minimum number of cycles when the dependence length is greater than the resource length; and

assigning the resource length as the minimum number of cycles when the resource length is greater than the dependence length.

11. (Currently amended) The method of claim 10, further comprising:

calculating the dependence length of the scheduling region based on the total height of a dependence graph of the scheduling region; and

calculating the resource length of the scheduling region based on the maximum number of cycles needed to schedule the one or more instructions ~~of~~ in the scheduling region for ~~a machine resource~~ the target processor.

12. (Canceled)

13. (Currently amended) An article comprising:

a computer readable medium having a plurality of machine accessible instructions stored thereon, which when executed by a computer, cause the computer to perform the following method:

for each of one or more instructions of one or more instruction types ready to be scheduled in a given cycle in a scheduling region, determining a new slack value based on ~~resource constraints, for each of one or more ready instructions in a scheduling region~~ a current maximum number of the instructions that can be scheduled in the given cycle for a target processor;

selecting ~~one of the ready instructions~~ up to the current maximum number of instructions from those instructions ready to be scheduled in the given cycle, based on a priority order associated with the new slack value ~~of the one ready instruction~~; and

scheduling the selected ~~ready~~ instructions~~; and~~

~~repeating the method for determining, selecting and scheduling for each of the one or more ready instructions remaining to be selected and scheduled until all ready instructions have been scheduled~~.

14. (Canceled)

15. (Currently amended) The medium of claim 13, wherein determining the new slack value comprises:

determining the new slack value ~~for each of the one or more ready instructions~~ based additionally on resource constraints and dependence height for each of the one or more instructions.

16. (Currently amended) The medium of claim ~~13~~ 15, wherein determining the new slack value comprises:

determining a dependence deadline based on [[a]] the dependence height for each of the one or more ~~ready~~ instructions;

determining a resource deadline based on the resource constraints for each of the one or more ~~ready~~ instructions;

selecting as a deadline value that indicates a least number of cycles, between the resource deadline and the dependence deadline; and

determining the new slack value based on the selected deadline value.

17. (Currently amended) The medium of claim 13, wherein ~~selecting one of the ready instructions~~ the priority order comprises ~~selecting a ready~~ an ordering of the one or more instructions based on their new slack value, wherein the instructions with lowest slack value ~~having a~~ have highest scheduling priority.

18. (Currently amended) The medium of claim 13, further comprising:

generating an entry in a ready list for each of the one or more ~~ready~~ instructions prior to determining the new slack value; and

removing the entry for the selected ~~ready~~ instructions from the ready list responsive to scheduling the selected instructions.

19. (Currently amended) The medium of claim 18, further comprising:

adding to an uncover list, any ~~non-ready~~ instructions not ready to be scheduled in the given cycle in the scheduling region that are uncovered by the scheduling of the selected ~~ready~~ instructions, wherein the ~~non-ready~~ instructions not ready to be scheduled are dependent on the selected instructions.

20. (Currently amended) The medium of claim 18, wherein the given cycle is a given clock cycle, further comprising:

advancing a virtual clock to a subsequent clock cycle when there are no ~~ready~~ instructions in the ready list that can be scheduled in [[a]] the given clock cycle; and

adding an entry to the ready list for any ~~non-ready~~ instruction not ready to be scheduled in the given cycle in the scheduling region that becomes ready in the subsequent clock cycle.

21. (Currently amended) The medium of claim 16, wherein determining the new slack value comprises:

determining a minimum number of cycles needed to schedule each of the one or more ~~ready~~ instructions in the scheduling region, taking the resource constraints into account;

determining the dependence deadline based on the dependence height and the minimum number of cycles; and

determining the resource deadline based on the resource constraints and the minimum number of cycles.

22. (Previously presented) The medium of claim 21, further wherein determining the minimum number of cycles comprises:

determining a dependence length of the scheduling region;

determining a resource length of the scheduling region;

assigning the dependence length as the minimum number of cycles when the dependence length is greater than the resource length; and

assigning the resource length as the minimum number of cycles when the resource length is greater than the dependence length.

23. (Currently amended) The medium of claim 22, further comprising:

calculating the dependence length of the scheduling region based on the total height of a dependence graph of the scheduling region; and

calculating the resource length of the scheduling region based on the maximum number of cycles needed to schedule the one or more instructions ~~of~~ in the scheduling region for ~~a machine resource~~ the target processor.

24. (Canceled)

25. (Currently amended) An apparatus for compiling a high-level programming language into an object code comprising:

a front end to receive a source code; and

a code generator, coupled to the front end, to:

receive the source code from the front end; and

compile the received source code into the object code,

wherein the code generator includes one or more resource-aware schedulers to:

for each of one or more instructions of one or more instruction types ready to be scheduled in a given cycle in a scheduling region, determine a new slack value based on ~~resource constraints, for each of one or more ready instructions in~~

~~a scheduling region~~ a current maximum number of the instructions that can be scheduled in the given cycle for a target processor;

select ~~one of the ready instructions~~ up to the current maximum number of instructions from those instructions ready to be scheduled in the given cycle, based on a priority order associated with the new slack value ~~of the one ready instruction~~; and

schedule the selected ~~ready~~ instructions~~; and~~

~~repeat the method for determining, selecting and scheduling for each of the one or more ready instructions remaining to be selected and scheduled until all ready instructions have been scheduled~~.

26. (Currently amended) The apparatus of claim 25, wherein the one or more resource-aware schedulers are to:

determine a first scheduling deadline for each of the one or more ~~ready~~ instructions in the scheduling region, taking dependence considerations into account;

determine a second scheduling deadline for each of the one or more ~~ready~~ instructions, taking resource constraints into account; and

select as a scheduling priority for each of the one or more ~~ready~~ instructions, between the first and second scheduling deadlines.

27. (Canceled)

28. (Currently amended) The apparatus of claim 26, wherein the priority order of the

one or more resource-aware schedulers ~~are to select the instruction for scheduling based on its scheduling priority~~ comprises an ordering of the one or more instructions based on their new slack value, wherein the instructions with lowest slack value have highest priority.

29-30. (Canceled)

31. (Currently amended) The apparatus of claim 25, wherein the one or more resource-aware schedulers are to schedule the one or more instructions such that instructions of a particular instruction type are distributed evenly among two or more resources.

32. (Currently amended) A system comprising:

a processor to execute each of one or more ready instructions; and

a memory system, coupled to the processor, to store each of the one or more ready instructions;

wherein the instructions include a resource-aware scheduler to:

for each of one or more instructions of one or more instruction types ready to be scheduled in a given cycle in a scheduling region, determine a new slack value based on ~~resource constraints, for each of one or more ready instructions in a scheduling region~~ a current maximum number of the instructions that can be scheduled in the given cycle for a target processor;

select ~~one of the ready instructions~~ up to the current maximum number of instructions from those instructions ready to be scheduled in the given cycle, based on a priority order associated with the new slack value ~~of the one ready instruction~~; and

schedule the selected ~~ready~~ instructions~~; and~~

~~repeat the method for determining, selecting and scheduling for each of the one or more ready instructions remaining to be selected and scheduled until all ready instructions have been scheduled~~.

33. (Previously presented) The system of claim 32, wherein the memory system comprises a Dynamic Random Access Memory (DRAM).

34. (Currently amended) The system of claim 32, wherein the resource-aware scheduler is to:

determine a first scheduling deadline for each of the one or more ~~ready~~ instructions in the scheduling region, taking dependence considerations into account;

determine a second scheduling deadline for each of the one or more ~~ready~~ instructions, taking resource constraints into account; and

select a scheduling priority for the instruction, between the first and second scheduling deadlines.

35. (Canceled)

36. (Currently amended) The system of claim 34, wherein the priority order of the resource-aware scheduler ~~is to select the instruction for scheduling based on its scheduling priority~~ comprises an ordering of the one or more instructions based on their new slack value, wherein the instructions with lowest slack value have highest priority.

37-38. (Canceled)